AIO Provider

CONTEC AIO Board

Version 1.0.1

User's Guide

July 17, 2012

[Remarks]

[Revision history]

Version	Date	Content
1.0.0.0	2011-7-12	First edition.
1.0.1.0	2012-5-29	Meta mode was added.
1.0.1	2012-7-17	Version rule of the document was changed.

[Hardware]

Model	Version	Notes
AIO-160802L-LPE		
ADI16-4(FIT)		Connection to I/O control module equipped with USB (CPU-CA10 (USB)) required.
DAI16-4(FIT)		Connection to I/O control module equipped with USB (CPU-CA10 (USB)) required.
ADI12-16(PCI)		
AIO121601E3-PE		

Contents

1. Introduction 4

2. Outline of Provider 5

2.1. Outline 5
2.2. Methods and properties 6
2.2.1. CaoWorkspace::AddController method 6
2.2.2. CaoController::OnMessage event 7
2.2.3. CaoController::AddVariable method 7
2.2.4. CaoController::get_VariableNames property 7
2.2.5. CaoVariable::get_Value property 7
2.2.6. CaoVariable::put_Value property 7
2.3. Variable list 8
2.3.1. Controller class 8
2.4. Error code 9
2.5. CAO-AIO API reference table 10

3. Sample Program 11

1. Introduction

This document is a user's guide of the AIO provider which is used to access CONTEC AIO board.
Refer to CONTEC API-AIO (WDM) Help (for PCI) or API-USBP (W32) Help (for USB) for details.

NOTE: The AIO device driver of the AIO board needs to be installed to use the AIO provider. Install the driver from API-PAC (W32) for PCI board or from API-USBP (WDM) for USB. After installing it, register the provider in the registry with reference to Table 2-1.

2. Outline of Provider

2.1. Outline

The AIO provider executes CONTEC API corresponding to CAO API at the time the CAO API is executed. Refer to Table 2-5 for CAO API and corresponding CONTEC API.

Table 2-1 AIO provider

File name	CaoProvAIO.dll
ProgID	CaoProv.CONTEC.AIO
Registry registration[1]	regsvr32 CaoProvAIO.dll
Remove registry registration	regsvr32 /u CaoProvAIO.dll

[1] The AIO board driver must be installed to register the AIO provider.

2.2. Methods and properties

2.2.1. CaoWorkspace::AddController method

The AIO provider establishes (opens) connection to the AIO board when the Controller object is created.

Syntax AddController(<bstrCtrlName:BSTR>, <bstrProvName:BSTR>, <bstrPcName:BSTR >
[, <bstrOption:BSTR>])

bstrCtrlName : [in] Controller name
bstrProvName : [in] Provider name (Fixed to "CaoProv.CONTEC.AIO")
bstrPcName : [in] Provider execution machine name
bstrOption : [in] Option character string

The machine name can be an empty string.

Following is a list of option string items.

Table 2-2 Option character string of CaoWorkspace::AddController

Option	Meaning
DeviceName=[<Device name>]	Device name of the board to be connected[*1] Default: "" (no value specified) For the case of "" (no value specified), connection to the first detected available device will be established. Note: Specify the device supporting the AIO board ID.[*1]
ScanCount=[<Number of retries>]	Number of retries (Range: 0 to 32767) Default: 4 times Number of retries performed when connection to the detected device fails for the case DeviceName option is set to "" (no value specified).
Interval=[<Sampling interval>][*2]	Sampling interval (Range: 0 to 65535) Default: 0 (off) Specify the sampling interval (ms) to acquire OnMessage event when the digital input[*1] byte changes.
Mask=[<Mask value>][*2]	Mask value (Range: 0 to 255) Default: 255 (no mask) Reduce occurrence of unnecessary events by masking the input byte when the Interval option is enabled.

*1: Refer to API-AIO (WDM) Help (for PCI) or API-USBP (W32) Help (for USB) for details.
*2: This is enabled only on the models equipped with digital input. Refer to CONTEC product manual for details.

2.2.2. CaoController::OnMessage event

Exchanges data with a client as an OnMessage event of CaoController class. At this time, received data is stored directly in Message::Value property.

Event number	Value	Meaning
1	Digital input byte	For the case the Interval option is set to other than off (0), this event occurs when the bit, which is not masked by the Mask option, changes at the time the digital input[*1*2] byte changes.

*1: Refer to API-AIO (WDM) Help (for PCI) or API-USBP (W32) Help (for USB) for details.
*2: This is available only on the models equipped with digital input. Refer to CONTEC product manual for details.

2.2.3. CaoController::AddVariable method

This method creates a variable object used to access the AIO board.
Only the variables given in 2.3.1 can be used. If a variable other than those is specified, this method will return an error.

Syntax AddVariable(<bstrName:BSTR > [, <bstrOption:BSTR>])

bstrName : [in] Arbitrary name
bstrOption : [in] Option character string (not used)

2.2.4. CaoController::get_VariableNames property

Acquires the variable name list in 2.3.1.

2.2.5. CaoVariable::get_Value property

Acquires information corresponding to a variable. For the implementation status and acquired data of each variable, refer to 2.3.1.

2.2.6. CaoVariable::put_Value property

Configures information corresponding to a variable. For the implementation status and setting data of each variable, refer to 2.3.1.

2.3. Variable list

2.3.1. Controller class

Table 2-3 Controller class user variable list

Variable name	Data type	Explanation	Attribute	
			get	put
AI? *2	VT_R4	Acquire a voltage value*1 of analog input CH?. Specify the logical number*6 after the variable name. Example: "AI1"	√	—
AO? *3	VT_R4	Output a specified voltage value*1 to analog output CH?. Specify the logical number*6 after the variable name. Example: "AO1"	—	√
DI? *4	VT_I2	Acquire a bit value (0 or 1)*1 of digital input bit?. Specify the logical number*6 after the variable name. Example: "DI1"	√	—
DO? *5	VT_I2	Output a bit value (0 or 1)*1 to digital output bit?. Specify the logical number*6 after the variable name. Example: "DO1"	—	√
DIB? *4	VT_I2	Acquire a byte value (0 to 255)*1 of digital input byte?. Specify the logical number*6 after the variable name. Example: "DIB1"	√	—
DOB? *5	VT_I2	Output a byte value (0 to 255)*1 to digital output byte?. Specify the logical number*6 after the variable name. Example: "DOB1"	—	√

*1: Refer to API-AIO (WDM) Help (for PCI) or API-USBP (W32) Help (for USB) for details.

*2: This is available only on the models equipped with analog input. Refer to CONTEC product manual for details.

*3: This is available only on the models equipped with analog output. Refer to CONTEC product manual for details.

*4: This is available only on the models equipped with digital input. Refer to CONTEC product manual for details.

*5: This is available only on the models equipped with digital output. Refer to CONTEC product manual for details.

*6: A variable object can be created with a logical number within the range of 0 to 99; however, the actual range for data acquisition/setting depends on the number of CHs implemented, etc. of the mounted model. Refer to CONTEC product manual for details.

Table 2-4 Controller class system variable list

Variable name	Data type	Explanation	Attribute	
			get	put
@MAX_AI[*2]	VT_I2	Acquire the maximum number of analog input channels[*1].	√	—
@MAX_AO[*3]	VT_I2	Acquire the maximum number of analog output channels[*1].	√	—
@ProcessId	VT_I4	Acquire process ID.	√	—
@DeviceName	VT_BSTR	Acquire device name[*1] of connected board.	√	—

*1: Refer to API-AIO (WDM) Help (for PCI) or API-USBP (W32) Help (for USB) for details.
*2: This is available only on the models equipped with analog input. Refer to CONTEC product manual for details.
*3: This is available only on the models equipped with analog output. Refer to CONTEC product manual for details.

2.4. Error code

The AIO provider does not have unique error codes.

For common errors of ORiN2, refer to the error code section in "ORiN2 Programming Guide".

2.5. CAO-AIO API reference table

The AIO provider executes API function that sets/acquires a value via CaoVariable.

Table 2-5 Controller class, variable class and corresponding AIO API

CAO API		AIO API*
Class::method name	Parameter/ variable name	
CaoWorkspace::AddController()	DeviceName	AioQueryDeviceName() AioInit() AioResetProcess() AioResetDevice()
CaoWorkspaces::Remove()	－	AioExit()
CaoVariable::get_Value()	AI?	AioSingleAiEx()
	DI?	AioInputDiBit()
	DIB?	AioInputDiByte()
	@MAX_AI	AioGetAiMaxChannels()
	@MAX_AO	AioGetAoMaxChannels()
	@ProcessId	－
	@DeviceName	－
CaoVariable::put_Value()	AO?	AioSingleAoEx()
	DO?	AioOutputDoBit()
	DOB?	AioOutputDoByte()

* Refer to CONTEC API-AIO (WDM) Help (for PCI) or API-USBP (W32) Help (for USB) for details of AIO API.

3. Sample Program

The following sample program shows the code to acquire AI CH1 voltage value with the variable "AI1".

List 3-1 SampleAi.frm

```
Private caoEng As CaoEngine
Private caoAIOl As CaoController
Private caoVar As CaoVariable

Private Sub Form_Load()

    Set caoEng = New CaoEngine
    Set caoAIOl = caoEng.Workspaces(0).AddController("SampleAi",  "CaoProv.CONTEC.AIO",  "",  _
                                                     "DeviceName=AIO001")
    Set caoVar = caoAIOl.AddVariable("AI1",    "")

End Sub


Private Sub cmdGet_Click()

    Dim sngRet As Single

    sngRet = caoVar.Value

    Text1.Text = CStr(sngRet)

End Sub
```

The following sample program shows the code to receive event at the time the digital input byte changes while the sampling interval is set to one second.

List 3-2 SampleEvent.frm

```
Private caoEng As CaoEngine
Private WithEvents caoAIOl As CaoController

Private Sub Form_Load()

    Set caoEng = New CaoEngine
    Set caoAIOl = caoEng.Workspaces(0).AddController("SampleEvent","CaoProv.CONTEC.AIO",  _
                                                     "","DeviceName=AIO001,Interval=1000")
End Sub


' Received event
Private Sub ctrl_OnMessage(ppCaoMess As CAOLib.ICaoMessage)

    ' Received digital input byte
    text2.Text = ppCaoMess.Value

End Sub60
```